02

Caracteres y string

Transcripción

[00:00] Hola. ¿Qué tal a todos? Sean bienvenidos a la quinta clase de su curso de Introducción a Java, primeros pasos. Continuando en donde estábamos en la clase anterior, a modo de repaso, vimos todo lo que es números enteros, tipo de dato para números enteros, para números decimales, y vimos algunos tipos de conversiones.

[00:24] Todo este código lo van a tener disponible en el repositorio. Ya está con algunas explicaciones aquí, para que quede un poco más claro. Y ahora vamos a ver otros tipos de dato, que en este caso son para guardar caracteres o palabras. Entonces, para recordar un poco vamos a crear una nueva clase y la vamos a llamar EjemploCaracteres. Le damos finish y ya tenemos nuestra nueva clase creada.

[00:56] En este caso normalmente iniciaríamos escribiendo nuestro típico método public static void main string, con arg de argumentos, pero no vamos a hacerlo en esta oportunidad porque les voy a enseñar un pequeño shortcut, una combinación de teclas para generar automáticamente este método y no estar escribiéndolo a mano a cada momento.

[01:22] Simplemente es escribir main, apretamos "Ctrl + espacio" y nos va a dar automáticamente la opción. Eclipse nos va a preguntar: "¿El método main que tú has escrito te refieres a este main?" Le damos que sí, vamos a apretar enter o darle doble clic, y él automáticamente va a generar el método por nosotros.

[01:45] Entonces llevo nuestro método aquí. Vamos a ver otro nuevo tipo de dato que es char. Como pueden ver, es una palabra reservada también de Java, vamos a llamar caracter a nuestra variable, y a diferencia de otros él solamente acepta un solo carácter o un solo número.

[02:07] Por ejemplo, a diferencia de string, nosotros no podemos definir un char vacío ni un char solamente a, porque con comilla doble nosotros definimos un valor string. Para definir un valor char, nosotros usamos comilla simples. De esa forma, si nosotros damos un System.out.println, de carácter, "Ctrl + espacio", seleccionamos, perfecto.

[02:53] Le damos aquí a rodar, le damos que sí si queremos guardar, porque no hemos salvado antes, y tenemos a en la consola. A diferencia de string, como ya les dije, caracter en este caso, si queremos reasignar el valor, él no va a soportar el valor vacío y tampoco va a soportar dos caracteres en uno solo. Él solamente soporta un caracter a la vez y también soporta números.

[03:27] Por ejemplo en este caso el número 1 hace referencia a un carácter de aquella tabla ASCII de caracteres donde tenemos pues letras mayúsculas, minúsculas, todos los números, notas musicales, etcétera. El número no hace referencia al número en sí, sino hace referencia al número de carácter de la tabla ASCII.

[03:53] Para los que no recuerdan, la tabla ASCII sería esta de aquí, en la cual tenemos toda la descripción de caracteres que existen. Hay una tabla ASCII moderna también, esta me parece que es la antigua, pero básicamente esta es la representación de char.

[04:15] Entonces, volviendo aquí, si yo pusiera por ejemplo 65, y doy un System.out.println, nuevamente de carácter, voy a salvar, voy a ejecutar, y él me da la letra A. ¿Por qué me da la letra A? Porque el 65 en aquella tabla ASCII, él representa a la letra A.

[04:52] Ahora, ¿qué pasaría si yo quisiera digamos hacer un 65 + 1? Él va a ejecutar la suma, porque esta suma de aquí va a dar un valor entero, digámoslo así, pero igual sigue representando un caracter. Otro ejercicio que quiero hacer acá es definir otra variable char, segundo caracter, y aquí vamos a llamar a caracter + 1. Y en este caso, a diferencia de aquí, él no está compilando.

[05:39] ¿Alguno de ustedes sabe más o menos por qué? Ya vimos esto en la clase anterior, pero a modo de refuerzo vamos a verlo aquí también, solamente para dejarlo bien en claro este tipo de conceptos de conversión. Número 1 es un valor tipo entero y caracter es un valor tipo char, que ya está definido aquí.

[06:05] Entonces, dado que en Java siempre que hacemos una suma de variables, va a mandar la variable más grande, él lo que está haciendo ahorita es convertir esto a un tipo entero, a un int. Pero nosotros estamos con una variable tipo char. Entonces esto que Eclipse nos va a decir aquí es simplemente que no puedo convertir de entero para char nuevamente.

[06:37] ¿Por qué? Porque a diferencia de aquí, donde estamos trabajando directamente con los valores y el valor calculado es un 66, que sería la letra B en todo caso, aquí estamos trabajando ya con la variable, y la variable ya está especificada como tipo char, a diferencia de aquí.

[06:57] Entonces, lo que tenemos que hacer aquí es forzar que sea un tipo char. Y tenemos que forzar que el resultado de esta suma sea tipo char. Entonces, como cualquier operación aritmética, aislamos esto, y así le decimos: "yo quiero que este resultado entre paréntesis, sea de tipo char".

[07:25] Nuevamente ejecutamos un System.out.println, vamos aquí, vamos a imprimir segundo caracter. "Ctrl + espacio", elegimos la alternativa, guardamos y ejecutamos. Y ya vemos bien claro que como caracter aquí aumentó en 1. Era 66. 66 + 1 = 67, que es la referencia a la letra C mayúscula de la tabla ASCII.

[08:00] Ahora, no es muy común trabajar solamente con caracteres, si nosotros necesitamos trabajar con palabras u oraciones, tenemos otro tipo de dato que

ya no es una palabra reservada de Java.

[08:15] Hasta ahora, lo que hemos estado viendo son tipos de variable llamados primitivos, por ejemplo en el caso de los enteros interés; en caso de decimales, double, y long para el caso de conversiones, y las demás variables como short, byte, son llamados variables primitivas.

[08:38] El caso de string es particular porque string no es una palabra reservada de Java ni es una variable primitiva. String es un objeto de Java. Más adelante, cuando ya veamos más de orientación a objetos, ya vamos a entender mucho mejor a este concepto.

[08:56] Por el momento no necesitamos aún entrar en profundidad sobre las cualidades que tiene string. Pero en este caso, por ejemplo string palabra, y aquí podemos escribir una variable como Alura cursos online. Si nosotros damos un System.out.println, vamos a ver que sería lo mismo como si nosotros pusiéramos esta palabra aquí adentro como parámetro. Guardamos, ejecutamos y tenemos Alura cursos online.

[09:48] De la misma forma que vimos anteriormente, si nosotros queremos concatenar strings usaríamos solamente el signo +. Entonces, si nosotros ponemos palabra, igual, palabra, + y el año actual que estamos 2020 y nuevamente damos un System.out.print, él va a crear una nueva string, un nuevo objeto string con el valor actualizado de las dos variables concatenadas. Tenemos aquí Alura cursos online y Alura cursos online 2020.